



Master thesis

**MSc Methodology of Biomedical Research,
Biostatistics and Clinical Bioinformatics**

University of Thessaly
School of Medicine

**Development of a software in Python for performing
independent t-test and one-way ANOVA with post-hoc test
considering Bonferonni's adjustment**

**Ανάπτυξη λογισμικού στην Python για την εκτέλεση ανεξάρτητου t-
test και one-way ANOVA με post-hoc test λαμβάνοντας υπόψη την
προσαρμογή του Bonferonni**

Three-member Advisory Committee:

Dr Axel Kowald (Supervisor)

Chrisoula Doxani

Professor Elias Zintzaras

Panagiotis Papadimitriou

Academic year:2016-2017

ABSTRACT

In the current diplomatic project is analysed basic statistical concepts as well as two statistical methods, well known as t-test and one-way ANOVA in order to test whether or not there is significant difference among the means of a number of groups. For each one presented the calculating methods and their usefulness. Additionally, based on their definitions and formulas we have developed algorithms that have the capability to apply the above methods with precision. Moreover, the formulation of these algorithms has been done using python and the examples derive from the field of biostatistics. The main purpose of this diplomatic project is the clear description and software development of descriptive statistics and the methods of, independent t-test and one-way ANOVA with post-hoc test considering Bonferroni's adjustment.

ΠΕΡΙΛΗΨΗ

Στην συγκεκριμένη διπλωματική εργασία, αναλύουμε βασικές στατιστικές έννοιες περιγραφικών χαρακτηριστικών καθώς και δύο στατιστικές μεθόδους, γνωστές ως t-test και one-way ANOVA για να ελέγξουμε εάν υπάρχει ή όχι σημαντική διαφορά μεταξύ των μέσων ενός αριθμού ομάδων. Για κάθε μια, παρουσιάσαμε τις μεθόδους υπολογισμού και τη χρησιμότητά τους. Επιπλέον, με βάση τους ορισμούς και τους τύπους τους έχουμε αναπτύξει αλγόριθμους που έχουν την ικανότητα να εφαρμόζουν τις παραπάνω μεθόδους με ακρίβεια. Επιπλέον, η διατύπωση αυτών των αλγορίθμων έχει γίνει χρησιμοποιώντας τη γλώσσα προγραμματισμού python και τα παραδείγματα προέρχονται από το πεδίο της βιοστατιστικής. Ο κύριος σκοπός αυτού του διπλωματικού έργου είναι η σαφής περιγραφή και ανάπτυξη λογισμικού προγράμματος για τα στατιστικά περιγραφικά χαρακτηριστικά και τις μεθόδους του ανεξάρτητου t-test και της one-way ANOVA με post-hoc test λαμβάνοντας υπόψη την προσαρμογή του Bonferroni.

CONTENTS

ABSTRACT

CHAPTER 1: INTRODUCE

1.1 Python programming language.....	1
1.2 Descriptive statistics.....	1
1.3 Program development for descriptive statistics.....	2
1.4 Program execution.....	4

CHAPTER 2: INDEPENDENT T-TEST

2.1 Independent t-test	6
2.2 Program development for independent T-test.	7
2.3 Program execution.....	10

CHAPTER 3: ONE-WAY ANOVA

3.1 One-way ANOVA with post-hoc test with Bonferonni's adjustment	12
3.2 Program development for One-way ANOVA with post-hoc test considering Bonferonni's adjustment.....	14
3.3 Program execution.....	18

CHAPTER 4: CONCLUSION

4.1 Methods summary and conclusions.....	21
--	----

CHAPTER 5: REFERENCES

5.1 References.....	22
---------------------	----

1.1 Python programming language

Python is a dynamic, interpreted (bytecode-compiled) language. There are no type declarations of variables, parameters, functions, or methods in source code. This makes the code short and flexible, and you lose the compile-time type checking of the source code. Python tracks the types of all values at runtime and flags code that does not make sense as it runs.

The interpreter throws, or "raises" in Python parlance, a runtime error if the code tries to read a variable that has not been assigned a value. Like C++ and Java, Python is case sensitive so "a" and "A" are different variables. The end of a line marks the end of a statement, so unlike C++ and Java, Python does not require a semicolon at the end of each statement. Comments begin with a '#' and extend to the end of the line.

The python programming language is useful in many fields one of this is biological/medical field. Python using statistics, bioinformatics and computational modelling for solve and investigate problems.

The main purpose of this thesis is using basic knowledge of python programming so we develop programs of statistics methods t-test for independent data and one-way ANOVA with post-hoc test considering Bonferonni's adjustment.

We will develop our Python programs in the Eclipse IDE (Integrated Development Environment), since this provides many helpful tools for writing and debugging programs.

1.2 Descriptive statistics

To start the programming of statistical methods, we will need to program some useful functions, these functions are the descriptive statistics.

Descriptive statistics are used to describe the basic features of the data in a study. They provide simple summaries about the sample and the measures. Together with simple graphics analysis, they form the basis of virtually every quantitative analysis of data. Are used to present quantitative descriptions in a manageable form. In a research study we may have lots of measures. Or we may measure a large number of data on any measure. Descriptive statistics help us to simplify large amounts of data in a sensible way. Each descriptive statistic reduces lots of data into a simpler summary.

The descriptive statistics we will use are following:

Mean

Simple or arithmetic average of a range of values or quantities, computed by dividing the total of all values by the number of values

Variance

The arithmetic mean of the squares of the deviations of all values in a set of numbers from their arithmetic mean. Variance and its square root (the standard deviation) are of fundamental importance as a measure of dispersion.

Standard deviation

Measure of the unpredictability of a random variable, expressed as the average deviation of a set of data from its arithmetic mean and computed as the positive square root of the variance.

Standard error

A standard error is the standard deviation of the sampling distribution of a statistic. Standard error is a statistical term that measures the accuracy with which a sample represents a population. In statistics, a sample mean deviates from the actual mean of a population; this deviation is the standard error.

Median

Value or quantity that falls halfway between a set of values arranged in an ascending or descending order. When the set contains an odd number of values, the median value is exactly in middle. If the number of values is even, then median is computed by averaging the two numbers closest to the middle.

Range

Measure of the variation in a set of data, computed by subtracting the lowest value from the highest value in the same set.

1.3 Program development for descriptive statistics

Opening the eclipse I create a project called Biostatistics, then in this project I create a python file with following title descriptive_statistics.py .

To start programming we need to call the math package into the program so that I can use various functions such as square root, absolute value etc.

So I start with the command:

```
import math
```

Then defining the definitions, we program the functions for descriptive statistics.

MAXIMUM:

```
# Maximum
def max(list):
    if len(list) > 0:      # If the list contains even a number
        m=list[0]         # Put on the variable with the first value of the list "list"
        for n in list:    # Run the list from start to finish
            if n>m:        #If the current list number is greater than m
                m=n        # make m the current number of the list
        return m          # returned m
    else:                 # else
        return 0          # returned 0
```

MINIMUM:

```
# Minimum
def min(list):
    if len(list) > 0:      # If the list contains even a number
        m=list[0]         # Put on the variable m the first value of the list "list"
        for n in list:    # Run the list from start to finish
            if n<m:        # If the current list number is less than m
                m=n         # make m the current number of the list
        return m          # returned m
    else:                  # else
        return 0           # returned 0
```

MEAN:

```
## mean
def mean(list):
    if len(list) > 0:
        _sum = 0.0         # initially we consider the average 0
        for n in list:     # Run the list from start to finish
            _sum = _sum + n # Put on _sum that was + the current number of the list
        return _sum/len(list) # Returns the average _sum / length of the list
    else:
        return 0
```

VARIANCE:

```
## Variance
def variance(list):
    m=mean(list)           # Put the average of the list first in the variable m
    v=0.0                  # Initialize v with 0
    for n in list:         # Run the list from start to finish
        v=v+pow(n-m, 2)    # Put on v the value that had + the force of difference of
                           # the current number in the list
                           # with the average value
    return v/(len(list)-1) # returned the value of the variance
```

STANDARD DEVIATION:

```
## Standard deviation
def sd(list):
    return math.sqrt(variance(list))
```


STANDARD ERROR:

```
## Standard Error = standard deviation / sqrt(n)
def se(list):
    if (len(list) > 0):
        return sd(list)/math.sqrt(len(list))
    else:
        return 0
```

MEDIAN:

```
## Median
def median(list):
    if len(list) > 0:
        list.sort()                # Sort the list
        if len(list) % 2 <> 0:      # number is even
            a=len(list)+1
            return list[(a-1)/2]
        else:                      # number is odd
            a=len(list)/2
            b=len(list)/2+1
            return (list[a-1]+list[b-1])/2
    else:
        return 0
```

RANGE:

```
## Range
def _range(list):
    return max(list)-min(list)
```

1.4 Program execution

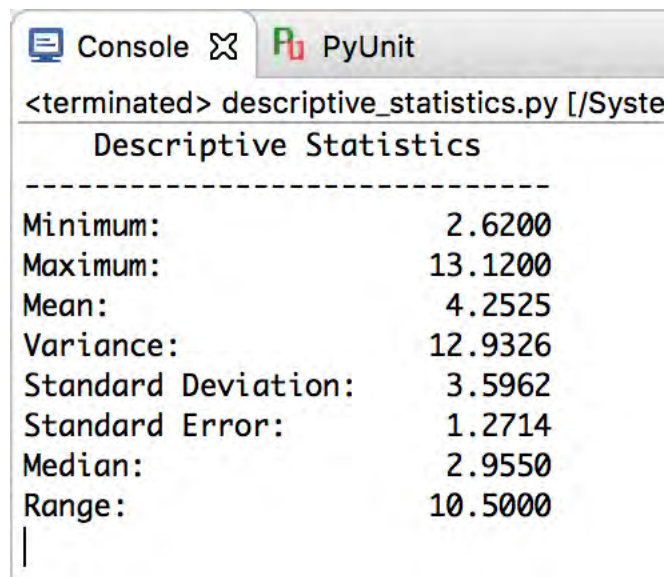
I create a new python file with the following title `descriptive_statistics_example.py` and then enter the following code:

```
from __future__ import division    # It allows me to score decimal in division 1/2,
                                   otherwise it gives me 0 .
from descriptive_statistics import * #With this way I call the functions of
                                   descriptive_statistics.py file
```

```
a = [2.75, 2.86, 3.37, 2.76, 2.62, 3.49, 3.05, 13.12] #example of list with data
```

```
print " Descriptive Statistics"  
print "-----"  
print "Minimum:      %10.4f"% min(a)  
print "Maximum:      %10.4f"% max(a)  
print "Mean:         %10.4f"% mean(a)  
print "Variance:     %10.4f"% variance(a)  
print "Standard Deviation: %10.4f"% sd(a)  
print "Standard Error:  %10.4f"% se(a)  
print "Median:       %10.4f"% median(a)  
print "Range:       %10.4f"% _range(a)
```

Then the console gives us the following results:



```
<terminated> descriptive_statistics.py [/Syste  
Descriptive Statistics  
-----  
Minimum:      2.6200  
Maximum:      13.1200  
Mean:         4.2525  
Variance:     12.9326  
Standard Deviation:  3.5962  
Standard Error:  1.2714  
Median:       2.9550  
Range:       10.5000  
|
```

INDEPENDENT T-TEST

2.1 Independent T-testT-test

T-test is a statistical test which is widely used to compare the mean of two groups of samples. It is therefore to evaluate whether the means of the two sets of data are statistically significantly different from each other.

There are many types of t-test :

The one-sample t-test, used to compare the mean of a population with a theoretical value.

The unpaired two sample t-test, used to compare the mean of two independent samples.

The paired t-test, used to compare the means between two related groups of samples.

We will study the t-test for independent samples

Independent t-test

The independent samples t-test compares the mean scores of two groups in a given variable, that is, two mean scores of the same variable, whereby one mean represents the average of that characteristic for one group and the other mean represents the average of that specific characteristic in the other group. Generally speaking, the independent samples t-test compares one measured characteristic between two groups of observations or measurements. It tells us whether the difference we see between the two independent samples is a true difference or whether it is just a random effect (statistical artifact) caused by skewed sampling.

Independent t-test formula

Let A and B represent the two groups to compare.

Let m_A and m_B represent the means of groups A and B, respectively.

Let n_A and n_B represent the sizes of group A and B, respectively.

The t test statistic value to test whether the means are different can be calculated as follow :

$$t = \frac{m_A - m_B}{\sqrt{\frac{S^2}{n_A} + \frac{S^2}{n_B}}}$$

S^2 is an estimator of the common variance of the two samples. It can be calculated as follow :

$$S^2 = \frac{\sum(x - m_A)^2 + \sum(x - m_B)^2}{n_A + n_B - 2}$$

Once t-test statistic value is determined, you have to read in t-test table the critical value of t distribution corresponding to the significance level alpha of your choice (5%) and compare them to each other. The degrees of freedom (df) used in this test are :

$$Df = n_A + n_B - 2$$

2.2 Program development for independent T-test

I create a python file with following title independent_t-test.py .

To start programming we need to call the following packages:

```
from __future__ import division      # It allows me to score decimal in division 1/2,
                                     otherwise it gives me 0
from descriptive_statistics import *  # this allows me to use the
                                     descriptive_statistics.py functions
from scipy.stats import t            # i will use this package for calculate the sig in
                                     conjunction with t and df
```

The following commands have to do with how the user input data and how corrects them in the event of incorrect import:

```
print 'Put the data for the first group and then type the "stop" word '
e=[]
i=1
key='start'
while key<>'stop' :
    print '%1d :'%i
    k=str(raw_input(' '))
    key=k
    if key<>'stop' :
        e.append(k)
    i=i+1
```

```

cor=1
while cor<>0 :
    print 'Check the data of the second group if some are incorrectly'
    print 'type their location and enter the correct values'
    print 'when all data is correct press 0 in the position to continue'
    print t
    cor=int(raw_input('position:'))
    _cor=cor-1
    if cor<>0 :
        e[_cor]=str(raw_input('value:'))
    for i in range(len(e)) :
        e[i]=float(e[i])
    print e

print 'Put the data for the second group and then type the "stop" word'
p=[]
i=1
key='start'
while key<>'stop' :
    print '%1d :'%i
    k=str(raw_input(' '))
    key=k
    if key<>'stop' :
        p.append(k)
    i=i+1

cor=1
while cor<>0 :
    print 'Check the data of the second group if some are incorrectly'
    print 'type their location and enter the correct values'
    print 'when all data is correct press 0 in the position to continue'
    print p
    cor=int(raw_input('position:'))
    _cor=cor-1
    if cor<>0 :
        p[_cor]=str(raw_input('value:'))
    for i in range(len(p)) :
        p[i]=float(p[i])
    print p

```

The t-distribution table for 0.05 level of Significance :

```
df_list={1: 12.71, 2:4.30, 3: 3.18, 4: 2.78, 5: 2.57, 6: 2.45, 7: 2.36, 8: 2.31, 9:2.26, \
10:2.23, 11:2.20, 12:2.18, 13: 2.16, \
14:2.14, 15:2.13, 16: 2.12, 17:2.11, 18: 2.10, 19:2.09, 20:2.09, 21:2.08, 22:2.07, \
23:2.07 ,24:2.06, 25:2.06 ,26:2.06 ,27:2.05, 28:2.05 ,29:2.05 ,30:2.04 }
```

Finally i enter the descriptive statistics commands, the commands to calculate the t-statistic and the commands for the t-test result:

```
print "Summary Statistics"
print "-----"
print "Group A"
print "n   : %10.4f"%len(e)
print "Mean : %10.4f"%mean(e)
print "SD   : %10.4f"%sd(e)
print "SE   : %10.4f"%se(e)
print "-----"
print "Group B"
print "-----"
print "n   : %10.4f"%len(p)
print "Mean : %10.4f"%mean(p)
print "SD   : %10.4f"%sd(p)
print "SE   : %10.4f"%se(p)
print "-----"
print "T-Test for independent data"
print "-----"
se_square_pooled=((len(e)-1)*pow(sd(e),2)+(len(p)-1)*pow(sd(p),2))/((len(e)-1)+(len(p)-1))
se=math.sqrt(se_square_pooled*(1/len(e)+1/len(p)))
_t=abs((mean(e)-mean(p))/se) # value of t
df=(len(e)-1)+len(p)-1      # degrees of freedom
sig=2*t.sf(_t, df)
print 'We answer the question by comparing the value t=%5.2f'%(_t)
print 'The value 5%% point of the t-distribution with %d df is %5.2f, so'%(df, df_list[df])
if (_t>df_list[df]):
    print "There is a significant difference between the means with p-value= %2.5f < 0.05" %sig
else:
    print "There is no significant difference between the means with p-value= %2.5f > 0.05" %sig
```

2.3 Program execution

We will examine with the help of the program one example of advance statistics courses of Professor Elias Zintzaras. The example represents a problem of a clinical study who compares the effectiveness of two treatments:


To assess the effectiveness of Tysabri, 24 patients with RRMS were randomized to receive either Tysabri or placebo. The primary end-point was the ARR. The results were as follows:


Tysabri (n=12)	Placebo (n=12)
0.38	0.80
0.37	0.70
0.24	0.39
0.08	0.06
0.04	0.49
0.03	0.07
0.07	0.63
0.19	0.83
0.35	0.62
0.38	0.95
0.27	0.92
0.20	0.81

Is Tysabri more effective than Placebo; alternatively, is the average ARR of Tysabri different than Placebo?

By entering the data into the software I have the following results:

```
Console
<terminated> independent_t-test.py [/System/Library/Frameworks/Python.framework/Versions/2.7/bin/python2.7]
Put the data for the first group and then type the "stop" word
1 :
0.38
2 :
0.37
3 :
0.24
4 :
0.08
5 :
0.04
6 :
0.03
7 :
0.07
8 :
0.19
9 :
0.35
10 :
0.38
11 :
0.27
12 :
0.20
13 :
stop
Check the data of the first group if some are incorrectly
type their location and enter the correct values
when all data is correct press 0 in the position to continue
['0.38', '0.37', '0.24', '0.08', '0.04', '0.03', '0.07', '0.19', '0.35', '0.38', '0.27', '0.20']
position:0
[0.38, 0.37, 0.24, 0.08, 0.04, 0.03, 0.07, 0.19, 0.35, 0.38, 0.27, 0.2]
```

```
Console 
<terminated> independent_t-test.py [/System/Library/Frameworks/Python.framework/Versions/2.7/bin/python2.7]
Put the data for the second group and then type the "stop" word
1 :
  0.80
2 :
  0.70
3 :
  0.39
4 :
  0.06
5 :
  0.49
6 :
  0.07
7 :
  0.63
8 :
  0.83
9 :
  0.62
10 :
  0.95
11 :
  0.92
12 :
  0.81
13 :
  stop
Check the data of the second group if some are incorrectly
type their location and enter the correct values
when all data is correct press 0 in the position to continue
['0.80', '0.70', '0.39', '0.06', '0.49', '0.07', '0.63', '0.83', '0.62', '0.95', '0.92', '0.81']
position:3
value:0.390
Check the data of the second group if some are incorrectly
type their location and enter the correct values
when all data is correct press 0 in the position to continue
['0.80', '0.70', '0.390', '0.06', '0.49', '0.07', '0.63', '0.83', '0.62', '0.95', '0.92', '0.81']
position:0
[0.8, 0.7, 0.39, 0.06, 0.49, 0.07, 0.63, 0.83, 0.62, 0.95, 0.92, 0.81]
```

```
Console 
<terminated> independent_t-test.py [/System/Library/Frameworks/Python.framework/Versions/2.7/bin/python2.7]
Summary Statistics
-----
Group A
n      :    12.0000
Mean   :    0.2167
SD      :    0.1366
SE      :    0.0394
-----
Group B
n      :    12.0000
Mean   :    0.6058
SD      :    0.3017
SE      :    0.0871
-----
T-Test for independent data
-----
We answer the question by comparing the value t= 4.07
The value 5% point of the t-distribution with 22 df is 2.07, so
There is a significant difference between the means with p-value= 0.00051 < 0.05
```


3.1 One-way ANOVA with post-hoc test with Bonferonni's adjustment

One-way ANOVA

The one-way analysis of variance (ANOVA) is used to determine whether there are any statistically significant differences between the means of three or more independent (unrelated) groups. The one-way ANOVA compares the means between the groups you are interested in and determines whether any of those means are statistically significantly different from each other. Specifically, it tests the null hypothesis:

$$H_0: \mu_1 = \mu_2 = \mu_3 = \dots = \mu_k$$

where μ = group mean and k = number of groups. If, however, the one-way ANOVA returns a statistically significant result, we accept the alternative hypothesis (H_A), which is that there are at least two group means that are statistically significantly different from each other.

At this point, it is important to realize that the one-way ANOVA is an omnibus test statistic and cannot tell you which specific groups were statistically significantly different from each other, only that at least two groups were. To determine which specific groups differed from each other, you need to use a post hoc test.

One-way ANOVA formula

$$SS_{\text{total}} = \sum (x_{ij} - \bar{x})^2$$

$$SS_{\text{within}} = \sum (x_{ij} - \bar{x}_i)^2$$

$$SS_{\text{within}} = SS_{\text{total}} - SS_{\text{between}}$$

$$df_{\text{total}} = N - 1 \quad df_{\text{between}} = r - 1 \quad df_{\text{within}} = df_{\text{total}} - df_{\text{between}}$$

$$MS_{\text{within}} = \frac{SS_{\text{within}}}{df_{\text{within}}}$$

$$MS_{\text{between}} = \frac{SS_{\text{between}}}{df_{\text{between}}}$$

$$F = \frac{MS_{\text{between}}}{MS_{\text{within}}}$$

x_{ij} = the j observation of i group

\bar{x} = the total mean of all observations

\bar{x}_i = the mean of i group

r = number of groups

N = total number of observations (all groups)

n = number of observations in group

Once F statistic value is determined, you have to read the F distribution table the critical value of F distribution corresponding to the significance level alpha of your choice (5%) and compare them to each other. The degrees of freedom used in this test are df_{within} and $df_{between}$.

Post-hoc test with Bonferonni's adjustment

If the F-ratio is significant, indicating that a difference between means exists, then post-hoc analyses can be performed to uncover the source of the significance or, in other words, to determine which specific means are different. A significant F-test may occur unexpectedly, in which case specific comparisons between factor levels, or contrasts, may be conducted. These contrasts are referred to as post-hoc comparisons. The appropriate post-hoc analysis is dependent upon the number and type of comparisons planned. If specific comparisons are planned or hypothesized up front, then these contrasts are referred to as a priori comparisons.

In Multiple Hypothesis Testing, the Bonferroni correction is a method for probability thresholding to control the occurrence of false positives.

When deciding whether to accept or reject an individual null hypothesis, a probability threshold, α , is utilized to control the likelihood of false positives. In multiple hypothesis testing, an increased number of samples, n , in a given family increases the probability that false positives will arise within that family at the same probability threshold, α . Thus, the threshold, α , should be lowered to control the total number of false positives.

The Bonferroni correction controls the number of false positives arising in each family by using a probability threshold of α/n for each observation within the family. By guaranteeing that the probability of a test being accepted within a family is the same as or less than the probability of any individual test being accepted.

Post-hoc test with Bonferonni's adjustment formula

For post hoc test in each pair of groups must to calculate:

$$t = \frac{\bar{x}_A - \bar{x}_B}{\sqrt{MS_{within}(\frac{1}{n_A} + \frac{1}{n_B})}}$$

For bonferoni's adjustment must to alculate:

$$\text{Bonf.} = \frac{r(r-1)}{2}, \quad T_{\text{bon.}} = t * \text{Bonf.}$$

Once t_{bon} value is determined, you have to read in t-test table the critical value of t distribution corresponding to the significance level alpha of your choice (5%) and compare them to each other. The degrees of freedom (df) used in this test are df_{within} .

3.2 Program development for One-way ANOVA with post-hoc test considering Bonferonni's adjustment

I create a python file with following title independent_t-test.py .

To start programming we need to call the following packages:

```
from __future__ import division      # It allows me to score decimal in division 1/2,
                                     otherwise it gives me 0
from descriptive_statistics import *  # this allows me to use the
                                     descriptive_statistics.py functions
from scipy.stats import t            # i will use this package for calculate the sig in
                                     conjunction with t and df
```

The following commands have to do with how the user input data and how corrects them in the event of incorrect import:

```
grnum=int(raw_input('type the number of groups'))
gr=[]
for n in range(grnum) :
    print 'Put the data for the group %1d and then type the "stop" word ' %(n+1)
    l=[]
    i=1
    key='start'
    while key<>'stop' :
        print '%1d :'%i
        k=str(raw_input(' '))
        key=k
        if key<>'stop' :
            l.append(k)
            i=i+1

    cor=1
    while cor<>0 :
        print "Check the data of the group %1d if some are incorrectly ,"%(n+1)
        print "type their location and enter the correct values,"
        print " when all data is correct press 0 in the position to continue "
        print l
        cor=int(raw_input('position:'))
        _cor=cor-1
        if cor<>0 :
            l[_cor]=str(raw_input('value:'))
    for i in range(len(l)) :
        l[i]=float(l[i])
    print l
    gr.append(l)
print gr
```

Within groups sum of squares:

```
x=[]
n_total=0
ss_w=0
for i in range(len(gr)) :
    x.append(mean(gr[i]))
    for j in range(len(gr[i])) :
        n_total=n_total +1
        ss_w=ss_w + pow(x[i] - gr[i][j] ,2)
```

Total sum of squares:

```
x_total=sum(x)

x_total=x_total/grnum
ss_t=0
for i in range(len(gr)) :
    for j in range(len(gr[i])) :
        ss_t=ss_t + pow(x_total - gr[i][j] ,2)
```

Between groups sum of squares:

```
ss_b= ss_t - ss_w
```

Degrees of freedom:

```
df_t=n_total-1
df_b=grnum-1
df_w= df_t - df_b
```

Mean of squares:

```
ms_w=ss_w/df_w
ms_b=ss_b/df_b
```

F statistic value and F distribution table:

F=ms_b/ms_w

df_f=[[161.4476 ,199.5000 ,215.7073 , 224.5832 , 230.1619 , 233.9860 , 236.7684 , 238.8827 ,
240.5433 , 241.8817],\

[18.5128 ,19.0000 , 19.1643 ,19.2468 ,19.2964 ,19.3295 ,19.3532 ,19.3710 ,19.3848 ,19.3959],\

[10.128, 9.5521, 9.2766, 9.1172, 9.0135, 8.9406, 8.8867, 8.8452, 8.8123, 8.7855],\

[7.7086, 6.9443 , 6.5914, 6.3882, 6.2561, 6.1631, 6.0942, 6.0410, 5.9988, 5.9644],\

[6.6079, 5.7861 , 5.4095, 5.1922, 5.0503, 4.9503, 4.8759, 4.8183, 4.7725, 4.7351],\

[5.9874, 5.1433 , 4.7571, 4.5337, 4.3874, 4.2839, 4.2067, 4.1468, 4.0990, 4.0600],\

[5.5914, 4.7374 ,4.3468, 4.1203, 3.9715, 3.8660, 3.7870, 3.7257, 3.6767, 3.6365],\

[5.3177, 4.4590, 4.0662, 3.8379, 3.6875, 3.5806, 3.5005, 3.4381, 3.3881, 3.3472],\

[5.1174, 4.2565 , 3.8625, 3.6331, 3.4817, 3.3738, 3.2927, 3.2296, 3.1789, 3.1373],\

[4.9646, 4.1028 , 3.7083, 3.4780, 3.3258, 3.2172, 3.1355, 3.0717, 3.0204, 2.9782],\

[4.8443, 3.9823 , 3.5874, 3.3567, 3.2039, 3.0946, 3.0123, 2.9480, 2.8962, 2.8536],\

[4.7472, 3.8853 ,3.4903, 3.2592, 3.1059, 2.9961, 2.9134, 2.8486, 2.7964, 2.7534],\

[4.6672, 3.8056, 3.4105, 3.1791, 3.0254, 2.9153, 2.8321, 2.7669, 2.7144, 2.6710],\

[4.6001, 3.7389 , 3.3439, 3.1122, 2.9582, 2.8477, 2.7642, 2.6987, 2.6458, 2.6022],\

[4.5431, 3.6823 , 3.2874, 3.0556, 2.9013, 2.7905, 2.7066, 2.6408, 2.5876, 2.5437],\

[4.4940, 3.6337 , 3.2389, 3.0069, 2.8524, 2.7413, 2.6572, 2.5911, 2.5377, 2.4935],\

[4.4513, 3.5915 ,3.1968, 2.9647, 2.8100, 2.6987, 2.6143, 2.5480, 2.4943, 2.4499],\

[4.4139, 3.5546, 3.1599, 2.9277, 2.7729, 2.6613, 2.5767, 2.5102, 2.4563, 2.4117],\

[4.3807, 3.5219 , 3.1274, 2.8951, 2.7401, 2.6283, 2.5435, 2.4768, 2.4227, 2.3779],\

[4.3512, 3.4928 , 3.0984, 2.8661, 2.7109, 2.5990, 2.5140, 2.4471, 2.3928, 2.3479],\

[4.3248, 3.4668 , 3.0725, 2.8401, 2.6848, 2.5727, 2.4876, 2.4205, 2.3660, 2.3210],\

[4.3009, 3.4434 ,3.0491, 2.8167, 2.6613, 2.5491, 2.4638, 2.3965, 2.3419, 2.2967],\

[4.2793, 3.4221, 3.0280, 2.7955, 2.6400, 2.5277, 2.4422, 2.3748, 2.3201, 2.2747],\

[4.2597, 3.4028 , 3.0088, 2.7763, 2.6207, 2.5082, 2.4226, 2.3551, 2.3002, 2.2547],\

[4.2417, 3.3852 , 2.9912, 2.7587, 2.6030, 2.4904, 2.4047, 2.3371, 2.2821, 2.2365],\

[4.2252, 3.3690 , 2.9752, 2.7426, 2.5868, 2.4741, 2.3883, 2.3205, 2.2655, 2.2197],\

[4.2100, 3.3541 ,2.9604, 2.7278, 2.5719, 2.4591, 2.3732, 2.3053, 2.2501, 2.2043],\

[4.1960, 3.3404, 2.9467, 2.7141, 2.5581, 2.4453, 2.3593, 2.2913, 2.2360, 2.1900],\

[4.1830, 3.3277 , 2.9340, 2.7014, 2.5454, 2.4324, 2.3463, 2.2783, 2.2229, 2.1768],\

[4.1709, 3.3158 , 2.9223, 2.6896, 2.5336, 2.4205, 2.3343, 2.2662, 2.2107, 2.1646]]

Comparison of F with table value .And post-hoc test with bonferoni's adjustment:

```

if (df_b<11) & (df_w<31) :
    if F>df_f[df_w][df_b] :
        print "There is a significant difference between the means of %d groups" %grnum
        print "-----"
        print "because the difference in average values is statistically significant,"
        print "you can make post hoc test ie multiple comparisons between the groups."
        posthoc=int(raw_input('If you want to do this press 0 else press 1:'))
        if posthoc==0 :
            """
            POST HOC TEST
            """
            print"-----POST HOC TEST-----"
            print ""
            print ""
            bonf = grnum*(grnum-1)/2

            for i in range(len(gr)) :
                for j in range(i+1,len(gr),1) :
                    __se=math.sqrt(ms_w*(1/len(gr[i])+1/len(gr[j])))
                    __t=abs((mean(gr[i])-mean(gr[j])) / __se )
                    df_pht=df_w # Βαθμοί ελευθερίας
                    sig=2*t.sf(__t, df_pht)
                    print "for group %d and group %d" %(i+1,j+1)
                    print "-----"
                    if (sig*bonf)<=0.05 :
                        print "There is a significant difference between the means of"
                        print "group %d and group %d with p-value= %2.4f < 0.05 " %(i+1,j+1,sig*bonf)
                        print ""
                        print ""
                    else :
                        if (sig*bonf)>1.0 :
                            pval=1
                        else :
                            pval=sig*bonf
                        print "There is no significant difference between the means of"
                        print "group %d and group %d with p-value= %2.4f > 0.05 " %(i+1,j+1,pval)
                        print ""
                        print ""
                else:
                    print "There is no significant difference between the means"
            else :
                print "we need to enter more data into the f-distribution table"

```

3.3 Program execution

We will examine with the help of the program the following:

In one study, liver weight (x) (expressed as a percentage of body weight) of mice belonging to $k = 4$ groups fed with 4 different diets was recorded. We want to investigate whether there are systematic differences between the four groups, ie to compare the average values of the four groups.

a	b	c	d
3.42	3.17	3.34	3.64
3.96	3.63	3.72	3.93
3.87	3.38	3.81	3.77
4.19	3.47	3.66	4.18
3.58	3.39	3.55	4.21
3.76	3.41	3.51	3.88

We run the program and enter the data:

```
Console
<terminated> one-way ANOVA.py [/System/Library/Frameworks/Python.framework/Versions/2.7/bin/python2.7]
type the number of groups4
Put the data for the group 1 and then type the "stop" word
1 :
  3.42
2 :
  3.96
3 :
  3.87
4 :
  4.19
5 :
  3.58
6 :
  3.76
7 :
  stop
Check the data of the group 1 if some are incorrectly ,
type their location and enter the correct values,
when all data is correct press 0 in the position to continue
['3.42', '3.96', '3.87', '4.19', '3.58', '3.76']
position:0
[3.42, 3.96, 3.87, 4.19, 3.58, 3.76]
Put the data for the group 2 and then type the "stop" word
1 :
  3.17
2 :
  3.63
3 :
  3.38
4 :
  3.47
5 :
  3.39
6 :
  3.41
7 :
  stop
Check the data of the group 2 if some are incorrectly ,
type their location and enter the correct values,
when all data is correct press 0 in the position to continue
['3.17', '3.63', '3.38', '3.47', '3.39', '3.41']
position:0
[3.17, 3.63, 3.38, 3.47, 3.39, 3.41]
```

```

Console
<terminated> one-way ANOVA.py [/System/Library/Frameworks/Python.framework/Versions/2.7/bin/python2.7]
[3.17, 3.63, 3.38, 3.47, 3.39, 3.41]
Put the data for the group 3 and then type the "stop" word
1 :
  3.34
2 :
  3.72
3 :
  3.81
4 :
  3.66
5 :
  3.55
6 :
  3.51
7 :
  stop
Check the data of the group 3 if some are incorrectly ,
type their location and enter the correct values,
when all data is correct press 0 in the position to continue
['3.34', '3.72', '3.81', '3.66', '3.55', '3.51']
position:0
[3.34, 3.72, 3.81, 3.66, 3.55, 3.51]
Put the data for the group 4 and then type the "stop" word
1 :
  3.64
2 :
  3.93
3 :
  3.77
4 :
  4.18
5 :
  4.21
6 :
  3.88
7 :
  stop
Check the data of the group 4 if some are incorrectly ,
type their location and enter the correct values,
when all data is correct press 0 in the position to continue
['3.64', '3.93', '3.77', '4.18', '4.21', '3.88']
position:0
[3.64, 3.93, 3.77, 4.18, 4.21, 3.88]

```


Finally i have the following results:

```
Console [X] [System/Library/Frameworks/Python.framework/Versions/2.7/bin/python2.7]
[3.42, 3.96, 3.87, 4.19, 3.58, 3.76], [3.17, 3.63, 3.38, 3.47, 3.39, 3.41], [3.34, 3.72, 3.81, 3.66, 3.55, 3.51], [3.64, 3.93, 3.77, 4.18, 4.21, 3.88]]
7.21976303527
There is a significant difference between the means of 4 groups
-----
because the difference in average values is statistically significant,
you can make post hoc test ie multiple comparisons between the groups.
If you want to do this press 0 else press 1:0
-----POST HOC TEST-----

for group 1 and group 2
-----
There is a significant difference between the means of
group 1 and group 2 with p-value= 0.0267 < 0.05

for group 1 and group 3
-----
There is no significant difference between the means of
group 1 and group 3 with p-value= 0.7040 > 0.05

for group 1 and group 4
-----
There is no significant difference between the means of
group 1 and group 4 with p-value= 1.0000 > 0.05

for group 2 and group 3
-----
There is no significant difference between the means of
group 2 and group 3 with p-value= 0.7955 > 0.05

for group 2 and group 4
-----
There is a significant difference between the means of
group 2 and group 4 with p-value= 0.0019 < 0.05

for group 3 and group 4
-----
There is no significant difference between the means of
group 3 and group 4 with p-value= 0.0696 > 0.05
```

4.1 Methods summary and conclusions

In the first example at unit 1.4 we applied the methods-functions of the program at a random sample of population and it came back as a result the descriptive statistics of sample. Finally, the operation of algorithm was successful.

In the second example at unit 2.3 after we develop a code in relation to independent t-test we applied it in example from the field of biostatistics from the advance statistics courses of professor Elias Zintzaras. As to our example, we would like to examine the effectiveness of tysabri drug comparing with placebo. The results present the summary statistics for each one group as well as that there are statistical significant difference between the means of two groups with $p\text{-value}=0.00051<0.05$ ie the tysabri is effective drug. Finally, the algorithm run successful.

In the third example at unit 3.3 we develop a code in regard to one-way ANOVA, which has the capability to conduct a post-hoc test in the case of statistical significant difference among the means of groups. We applied this code in example from the advance statistics courses of professor Elias Zintzaras in witch we look in to whether or not there are statistical significant difference among four diets in four teams of mice. The results of one-way ANOVA presents this there are statistical significant difference among the means of liver weight of four teams. Moreover, we conducted individual comparisons using post-hoc test with bonferonni's adjustment. So the groups which present statistical significant difference between the mean of liver weight of mice of each group are, 1 – 2 with $p\text{-value}=0.0267<0.05$ and 2 – 4 with $p\text{-value}=0.0019<0.05$. Finally, the algorithm run successful.

5.1 References

1. Zinzaras, 2016-2017, Course Advanced_Statistics.
2. Axel Kowald ,2016-2017 , Course Computer programming and databases.
3. John Kleinberg ,Eva Tardos Algorithm Design.
4. Michael T. Goodrich ,Roberto Tamassia ,Michael H. Goldwasser ,
Data Structures and Algorithms in Python.
5. Takis Papaioannou ,Sotiris B. Loukas ,Introduction to statistics
second edition .
6. Takis Papaioannou , Kosmas Ferentinou , mathematical statistics
second edition .
7. P. D. Bozanis ,Data structures .